WebSphere® software

IBM

Redpaper

Dustin Amrhein
Ruth Willenborg

Adopting Cloud Computing using the WebSphere CloudBurst Appliance

Introduction

Are you interested in lowering costs by using existing hardware resources more efficiently? Do you want to increase efficiency with faster and more repeatable deployments of application environments? Most likely you are interested in these improvements, but where and how do you start?

This IBM® Redpaper™ publication describes the benefits of private cloud computing, using the WebSphere® CloudBurst™ Appliance (referred to as *WebSphere CloudBurst*).

WebSphere CloudBurst is used in the daily operations of the WebSphere Application Server Development and Test Organization (WASDTO), part of the IBM Software Group. Using WebSphere CloudBurst, WASDTO successfully demonstrated the benefits of private cloud computing. This paper defines the approach taken and the results achieved from the incremental adoption of WebSphere CloudBurst technology. As a result of this implementation, hardware utilization was increased from an average of 10% to an average of 60%, deployment times were shortened from 3 hours to 20 minutes, and security issues from non-compliant deployments were eliminated.

The WASDTO team is responsible for testing new releases of WebSphere Application Server, spanning eight releases and a test laboratory with over 2000 servers. The capabilities of the appliance were demonstrated by the team's implementation of many automated processes, including the following processes:

- Rapid deployment and removal of WebSphere topologies
- Intelligent management of deployments across hypervisors for optimum performance
- Customization of patterns to meet security policies

This paper also describes how the team approached the adoption of cloud computing, including business, organizational, and technological perspectives. Each discussion outlines the problem, the specific approach that was taken by WASDTO, and a plan to help you apply our experience to your own adoption of cloud computing.

WASDTO is in the process of gradually adopting a cloud computing approach for its test environments and has currently moved approximately 6% of its infrastructure into a WebSphere CloudBurst-managed environment. Already, the WASDTO team has realized USD .5 million in direct savings and USD 2.1 million from the adoption in agile process improvements. Although we cannot ensure that these results will occur with every adoption of WebSphere CloudBurst, this paper helps you to develop a staged adoption plan, gain organizational buy-in, develop an implementation plan, and measure your successes.

Adopting the WebSphere CloudBurst solution

Cloud computing promises to significantly affect the way that you interact with your middleware application environments. Gone are the days of waiting for weeks to acquire hardware, install software, and tweak configuration settings until everything is just right. By using cloud computing techniques, you can create fully configured application environments and deploy them on top of shared, virtualized infrastructure in minutes.

However, to take advantage of cloud computing, you need to acknowledge and plan for the disruptive nature of these types of solutions. As an example, adopting cloud computing typically results in the creation of new roles within teams and new interaction points between existing teams. Are you ready for this kind of change? By studying the adoption of WebSphere CloudBurst by WASDTO, you can learn how to mitigate these disruptions to ensure an effective adoption path.

The WebSphere Application Server Development and Test Organization (WASDTO)

What is the mission of WASDTO? This organization performs continuous WebSphere Application Server testing. Daily, the team deploys and tests WebSphere Application Server builds on multiple operating system platforms, using over 2000 physical servers in a laboratory setting.

To perform testing at this scale, WASDTO places a high priority on automation. In fact, substantial automation infrastructure existed prior to the adoption of WebSphere CloudBurst. This automation consists of several components:

- A custom-developed application for the automated leasing of hardware resources
- Tivoli® Provisioning Manager to automate the installation of operating systems to target machines
- Custom installation scripts to install the WebSphere Application Server product
- Custom configuration scripts to create the desired WebSphere Application Server topologies
- Custom configuration scripts to install and configure the applications that are used for testing

Combined, these components enable a highly automated process that governs the installation, configuration, and retrieval of an entire software stack (including WebSphere Application Server and the operating system) for testing. In fact, prior to adopting WebSphere CloudBurst, WASDTO was able to install and completely configure a single test environment in as little as 3 hours.

These automation components are still in use by WASDTO today, illustrating two key points about the adoption of cloud computing approaches, such as WebSphere CloudBurst:

- You need to have the freedom to incrementally adopt the solution, and you must be able to integrate the use of that solution into existing processes.
- As with WASDTO, an incremental, integrated adoption path provides an easier entry point for using the technology, while still delivering significant return on investment (ROI).

Motivations for adoption

With the presence of efficient automation in effect, what motivated WASDTO to turn to WebSphere CloudBurst? Put simply, the organization sought improvement in the areas of availability, utilization, and manageability:

- *Availability:* Automated provisioning of WebSphere Application Server environments using the team's traditional approaches produced a failure rate between 20-50%. In many cases, the complexity and the number of components involved in the provisioning process contributed to these failures. In all cases, these failures had an adverse effect on the team's throughput in terms of numbers of tests run.
- *Utilization:* Like many other organizations, especially those organizations involved in development and testing, under-utilization of physical servers was a serious concern. On average, utilization rates in the 2000+ server test lab were between 6-12%. With budgetary constraints preventing the procurement of additional servers, the team needed to find a way to better utilize existing resources.
- *Manageability:* WASDTO comprises a large number of small, agile teams. Members of these teams vary in hardware and software administration expertise. Thus, managing hardware and software resources effectively had been problematic.

In addition to improving in these areas, WASDTO established other requirements for the adoption of WebSphere CloudBurst in its environment. Namely, the organization wanted to minimize the costs of adoption by making use of existing assets. These assets included both physical hardware and software resources, such as scripts, applications, and tests.

These requirements matched well with the capabilities of WebSphere CloudBurst. WASDTO liked the design concept of the appliance, in that it allowed WASDTO to start using it with little setup. In addition, the Bring Your Own Cloud model, where users supply their own cloud resources (see Figure 2 on page 10), allows the reuse and consolidation of existing hardware. The command-line interface (CLI) and Representational State Transfer (REST) application program interfaces (APIs) provided the means to integrate WebSphere CloudBurst into WASDTO's existing processes. As well, WebSphere CloudBurst script packages allowed WASDTO to reuse existing WebSphere Application Server configuration scripts.

A plan for adoption

After establishing the focal areas for improvement, WASDTO decided to use WebSphere CloudBurst as a means to achieve that improvement. The leadership of WASDTO constructed a plan for adoption, during which the system was introduced into the environment incrementally. WASDTO used the following three major stages of adoption:

- *Stage 1:* Exploit WebSphere CloudBurst within existing provisioning processes
- *Stage 2:* Use WebSphere CloudBurst to provide self-service deployment capabilities for novice administrators of the product
- *Stage 3:* Expose the product to expert WebSphere Application Server administrators

The organization carefully selected these stages for the purpose of controlling the rollout of the appliance, while simultaneously targeting users and processes with pain points that best mapped to the capabilities of the appliance. In this way, adopting the appliance was minimally disruptive to day-to-day WASDTO operations.

Enacting a staged adoption approach

In the first stage of adoption, the team used the product CLI to achieve fast, consistent provisioning of WebSphere Application Server environments from within existing processes. In this way, the organization benefited from the improved availability and utilization provided by the product, but there was no disruption to the test users. Other than more rapidly available environments, the users noticed little difference in the processes for gaining access to WebSphere Application Server environments. This example highlights the seamless fashion by which the appliance integrates into existing provisioning processes.

In the second stage of adoption, WASDTO targeted improved manageability for a group of novice WebSphere CloudBurst administrators. WASDTO built a set of standard patterns that represented the environments needed by this group of people to conduct development and test activities. The group was granted access to deploy those patterns within the product. The WebSphere CloudBurst approach to provisioning solved major pain points for these less experienced administrators. The patterns encapsulated the necessary configuration for their environments, so they did not have to perform administration activities. They only needed to select a pattern, deploy it, and access was provided within minutes. Further, the users were confident that they were able to reliably reproduce any of their development and test environments with the push of a button.

For the third stage of adoption, WASDTO leadership targeted the seasoned WebSphere Application Server users and administrators. The group did not feel the same pain points as the less experienced administrators and, for this reason, were less inclined to contribute their hardware resources to WebSphere CloudBurst. However, WASDTO leadership used persuasive evidence, such as deployment times that were reduced from hours to minutes and significant decreases in deployment errors, to gradually get more of these users on-board.

Gaining organizational buy-in

While enacting the staged adoption approach for the appliance, WASDTO leaders also enacted a plan to gain organizational buy-in by quantitatively demonstrating the benefits of the appliance. The key to organizational buy-in for WASDTO was to clearly communicate the value at each stage of the adoption process and holding regularly scheduled meetings with stakeholders and decision makers throughout each stage of adoption. During adoption meetings, WASDTO provided data to validate its ongoing adoption efforts.

To provide this data, the team first identified metrics that were applicable to each stage of adoption. Then, the team proceeded to establish a method for measuring and recording these metrics for both WebSphere CloudBurst and traditional provisioning approaches. The metrics included key indicators, such as time to deploy, ratio of healthy deployments, ratio of compliant deployments, and more. After identifying generally applicable metrics, the team identified metrics unique to particular stages of adoption. For instance, for adoption stages two and three, the team gathered user satisfaction data.

Each time that WASDTO met with stakeholders, the WASDTO team conveyed up-to-date data relevant to that particular stage of adoption. This quantifiable data was an integral part of gaining buy-in.

In addition to regularly scheduled meetings with key decision makers, WASDTO also engaged other teams that it thought might benefit from using the appliance. The WASDTO team members acted as eager internal advocates for WebSphere CloudBurst by providing demonstrations and hands-on access to other teams. In effect, the team set about starting a grass roots movement in favor of spreading the usage of the appliance by proactively educating and preparing other potential adopters.

By focusing on buy-in from both a top-down and bottom-up perspective, WASDTO secured executive-level sponsorship for, and increased internal use of, WebSphere CloudBurst. This executive sponsorship was critical in the efforts to increase WASDTO cloud resources and increase the integration of the appliance in WASDTO. In addition, by increasing internal use of the appliance, WASDTO acted as a catalyst for the distribution of the benefits afforded by the appliance.

Your adoption plan

The adoption of WebSphere CloudBurst by WASDTO reinforces the need for an adoption plan. Your plan needs to include a set of problems to tackle within the enterprise. Plan your capacity needs as you move forward with an incremental adoption plan, based on results borne out by empirical data. Your adoption plan needs to focus on what barriers your organization or teams face in becoming more efficient. This plan must include how cloud and WebSphere CloudBurst can address these barriers by working with, and augmenting, your existing infrastructure solutions.

Identify meaningful metrics, and measure them for both traditional and WebSphere CloudBurst approaches. Convey these results to key decision makers within the organization, so that decision makers are clear about the value provided by cloud technology. Listen to the users, and focus on incrementally adding value to your WebSphere CloudBurst patterns based on their needs.

Infrastructure for the cloud

Cloud computing is a broadly used term; one in which the concepts apply to a number of elements within information technology (IT). For this reason, it is often helpful to categorize a cloud computing solution to clarify the types of services it provides. In general, cloud computing solutions can be categorized in the following ways:

- *Infrastructure as a Service (IaaS):* Providing physical compute resources, such as servers, storage, and networking infrastructure, as a set of services using cloud computing techniques
- *Platform as a Service (PaaS):* Providing application platforms, such as web servers, databases, and integration components, as a set of services using cloud computing techniques
- *Software as a Service (SaaS):* Providing user applications as services using cloud computing techniques

In addition to clarifying the type of a cloud computing solution to implement, it is also necessary to specify the delivery model:

- *Public clouds:* The service provider owns and manages services delivered by the cloud.
- *Private:* The enterprise owns and manages services delivered by the cloud.
- *Hybrid:* An environment that consumes both public and private cloud services.

In many cases, IaaS solutions provide a minimally disruptive entry point into cloud computing. These solutions focus on effectively virtualizing your physical hardware to enable a higher utilization rate, faster provisioning, and simplified resource management. When using IaaS solutions, your software components remain the same, but the mode of delivery changes (for example, virtual machines versus traditional installations). In many cases, you can use the existing skills and assets within the organization.

The delivery model of a particular cloud solution is an important factor in adoption as well. For instance, let us presume you are considering the adoption of an IaaS solution. If a service provider delivers the solution using a public cloud, it means that the service provider owns, manages, and controls the physical hardware that makes up the cloud. You sacrifice a certain level of control and insight so that you do not have to own the hardware elements of the cloud.

Alternatively, if you consider a private cloud solution, the physical resources that make up the cloud are on premise, within the organizational firewall. Although you need to procure resources for the cloud, you also benefit from a higher level of control and insight into the makeup of the cloud. In addition, certain IaaS solutions, including WebSphere CloudBurst, do not require a new capital outlay. They allow you to make better use of the resources that already exist within your organization.

Forming the cloud

WebSphere CloudBurst provides drop-in appliance technology to assist in quickly building a private IaaS cloud computing solution, either for direct self-service use, or as the foundation for your PaaS or SaaS solutions. It provides an effective way to virtualize and abstract existing physical infrastructure and to form a logical pool of resources used to host virtualized WebSphere application environments. Specifically, the WebSphere CloudBurst cloud consists of hypervisors (virtualization platforms, for example, VMware ESX, IBM PowerVM™, or IBM z/VM®, that allow multiple operating systems to run on a host), storage, and network infrastructure (a subnet with a pool of IP addresses). See Figure 1.

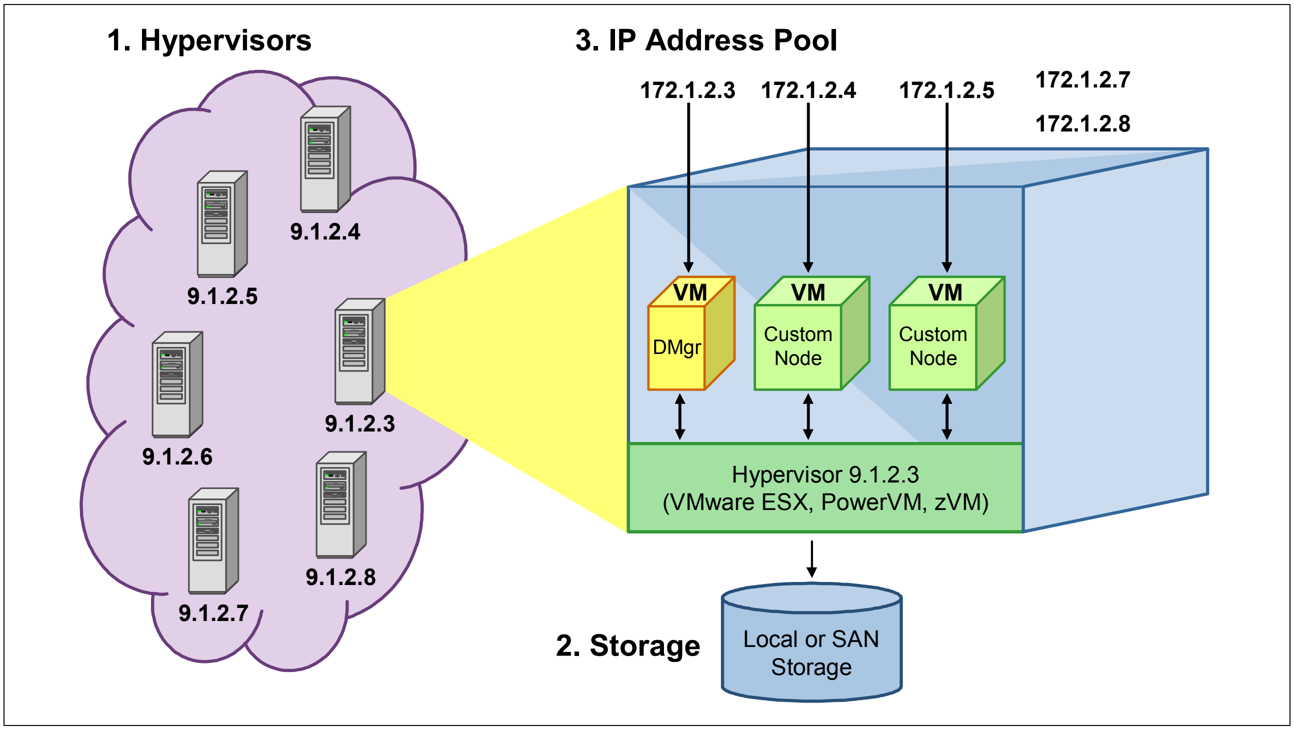

Figure 1 WebSphere CloudBurst cloud

The appliance operates on the Bring Your Own Cloud model, so users must separately obtain these resources for WebSphere CloudBurst. This approach affords the user a high degree of control over precisely which resources and the amount of those resources to dedicate to the cloud. It also enables organizations to more effectively use resources that already exist within the organization.

WASDTO staged the transitioning of resources into the appliance, corresponding to the needs of each adoption phase. In sizing the resource requirements, the team analyzed the CPU, memory, and storage requirements of the cloud by considering two major upper limits: planned unique virtual images and total virtual machines deployed.

Both the number of unique virtual images and total number of virtual machines deployed affected the storage requirements for the WASDTO cloud. When the appliance deploys a virtual image to a hypervisor for the first time, it creates a cache for the image. This cache significantly speeds up all subsequent deployments, and each virtual image has a unique cache. Naturally, this cache accounts for storage, so for each WebSphere Application Server Hypervisor Edition image, the team had to ensure that there was enough storage on the hypervisor hosts in the cloud. As a note, the recommended approach with WebSphere CloudBurst is to create a small set of distinct images and a larger set of patterns. This approach reduces storage requirements in a cloud and also prevents escalating management costs caused by too many virtual images.

The WASDTO team members also considered how many virtual machines they wanted to concurrently support in the cloud. This decision, too, affected storage requirements, because each virtual machine requires its own share of storage. In addition, the target number of concurrent machines helped to determine the amount of CPU and memory resources that the cloud required.

Considering these storage, CPU, and memory resource requirements, WASDTO determined the initial number of hypervisors needed for the cloud. Over time, the appliance grew to include up to 6% of the resources in the WASDTO lab.

It is also important to note the type of physical servers and storage that WASDTO used. The team settled on mostly commodity hardware, each with between 1 and 2 gigabytes (GB) of memory. In addition, the team settled on the use of local storage over that of storage area network (SAN)-based storage. For each core on a server, the team configured 20 GB of available storage (the amount required by their virtual machines). This configuration of machines provided the team with benefits in two key areas:

- *Cost:* The team was able to reuse existing, commodity hardware, thus requiring no new capital outlay. In addition, the use of local storage over a more expensive SAN-based configuration contributed to keeping costs down.
- *Performance:* After performing tests using both local and SAN-based storage approaches, WASDTO noticed that after the creation of the image cache, local storage provided consistently better deployment times. The team attributed this result to the fact that there was less input/output (I/O) contention when concurrently deploying several virtual machines using local storage as opposed to SAN storage. When using local storage, fewer virtual machines needed concurrent access to the same set of virtual image files. However, when using SAN storage, it was possible, and in fact likely, that multiple virtual machines required concurrent access to the single set of virtual image files stored on the SAN. Therefore, I/O contention on the SAN increased, which slowed deployment. Note that, even though WASDTO observed these performance results, you must perform your own testing based on your expected concurrency to determine the best storage options in your cloud.

In addition to determining the number of servers and amount of storage for the cloud, WASDTO also set up the necessary network infrastructure. First, the team members determined the number of required IP addresses for their cloud by considering their target number of concurrent virtual machine deployments. Each virtual machine requires a unique IP address, so the minimum number of addresses required was equal to the target number of concurrent deployments. After that, the team members collaborated with the network administration team to gather information about the subnet that their virtual machines were to use. This information included addresses for the subnet, gateway, and Domain Name Servers (DNSs).

After the team identified the resources for the cloud, it was able to define cloud groups, hypervisors, and IP groups in WebSphere CloudBurst. With the cloud defined, the team was ready to start deploying patterns to the cloud.

Managing the cloud

One key benefit of WebSphere CloudBurst is that it drives operational efficiency in the process of creating, deploying, and managing WebSphere application environments. WASDTO benefited immediately from commercial-off-the-shelf (COTS) enhancements that were provided by the appliance. These enhancements included more consistent deployment configurations and significantly reduced deployment times. Beyond these COTS enhancements, the team continued to evolve and refine the way that it used the appliance to produce even better results. The team further automated the cloud management and used the cloud resource controls that were provided by the appliance.

Automated cloud management

Further automation efforts included enabling an elastic cloud, improving monitoring and governance, and using the existing resource pools better.

Priming the cache

Although the appliance delivered significantly improved deployment times, WASDTO wanted to be able to predetermine deployment times. In WebSphere CloudBurst, during the first deployment of a virtual image to a particular hypervisor, the appliance transfers the entire image from the appliance to the hypervisor. At that point, the appliance creates an image cache on the hypervisor host. Subsequent deployments do not require the transfer of the entire image, and thus, they are considerably faster.

WASDTO wanted to ensure that users consistently experienced fast deployment times. For this reason, each time that WASDTO imports a new version of WebSphere Application Server Hypervisor Edition into the appliance, it initiates a CLI script that automates the priming of the hypervisor cache for the new image. The script creates a pattern using the new image and forces a deployment to each possible disk storage system for the new image.

Expanding and contracting the cloud

Over time, it became apparent that there were peaks and valleys in using WebSphere CloudBurst. The team decided that during peak usage, it might spare additional resources from the test lab for the WebSphere CloudBurst cloud. During off-peak times, the team can return those resources to the lab pool, making the resources available for other purposes.

WASDTO uses a combination of IBM Tivoli Provisioning Manager and the WebSphere CloudBurst CLI to enable an elastic cloud. We discuss this concept in “Automated management for cloud infrastructure” on page 17.

Monitoring and governance in the cloud

The product CLI provides an excellent means to manage and monitor activity within the WebSphere CloudBurst cloud. For example, because security status must be reevaluated after a 6 week period, WASDTO created a simple script that identifies owners of virtual systems exceeding this period. This script helps to quickly create the necessary security audit reports, which, in turn, are used to contact the appropriate owners so that they can take action on their virtual systems.

Cloud resource controls

As more users and teams began to use WebSphere CloudBurst, the demand for cloud resources grew, and at the same time, new usage scenarios for the cloud began to emerge. To support the growing number of users and the emerging usage scenarios, WASDTO decided the best approach was to divide the WebSphere CloudBurst cloud into multiple resource pools.

The team uses WebSphere CloudBurst cloud groups to logically group sets of hypervisors into deployment targets. In this way, the team can create purpose-built subclouds that contain the resources to support special use scenarios, such as performance and stress testing. Additionally, the team controls access to cloud groups at the user and group levels, while maintaining centralized management capabilities using a single appliance. This approach allows the team to effectively distribute cloud resources and prevent any one team or project from consuming all of the cloud resources.

Planning for your cloud

The way in which WASDTO set up and refined the use of the WebSphere CloudBurst cloud provides a model worth imitating. First, it is important to identify deployment goals in using the product. Second, with these goals identified, proceed by sizing the initial amount of resource that your cloud requires. With the sizing information in place, identify the teams that will provide you with the servers, storage, and networking components required. Engaging these teams as soon as possible is critical to avoiding barriers that slow the time to implementation.

After you procure your cloud resources and define them in WebSphere CloudBurst, you can start to use the appliance to deploy product patterns. You will benefit from the COTS capabilities of the appliance, but like WASDTO, be cognizant of further areas for operational improvement in your organization. Like WASDTO, employ automation and use resource controls where appropriate, because automation and resource controls will improve your operational efficiency.

Security and compliance

Cloud computing proposes a new model for service delivery, and along with that new model comes new security and compliance considerations. For instance, the concepts of self-service access and shared resource pools demand attention from a security perspective. Effective self-service access and resource sharing are only possible when coupled with the ability to define user permissions and resource access.

Further, although you deliver services in a new manner, often using virtualization, you still have organizational compliance standards to meet. The speed and simplicity of creating new virtual services add additional pressure to compliance processes. Any services, whether delivered using traditional means or with the cloud, must adhere to the same set of standards.

WASDTO approached these facets of security and compliance in implementing its WebSphere CloudBurst cloud. The team members addressed the issue of securing the environments that were deployed by the appliance, and they created a secure approach to allow users to access and utilize resources.

Securing environments in the cloud

As is typical throughout many enterprises, IBM has strict compliance rules for any operating system environments in the testing laboratories. Compliance rules include required software levels and an extensive list of the latest security patches. Traditionally, these requirements caused WASDTO difficulty when using virtual images as a means to deploy environments.

Prior to adopting WebSphere CloudBurst, the virtual images produced by the test organization captured an operating system of a particular version and set of patches. Typically, the images stayed in use for an extended period, which meant that, not long after creating the image, the virtual machines it produced were no longer compliant. Manually updating these virtual machines was time-consuming and error prone. Eventually, so that it satisfied compliance requirements, WASDTO created a completely segregated and firewalled network to host its virtual machines. This level of isolation satisfied compliance and security concerns, but these firewalled networks made it more difficult for developers and testers to access testing environments.

With WebSphere CloudBurst and the WebSphere Application Server Hypervisor Edition virtual images, WASDTO was able to overcome these historical challenges. To start, the team uses the extend and capture mechanism that is provided by the product, as shown in Figure 2.

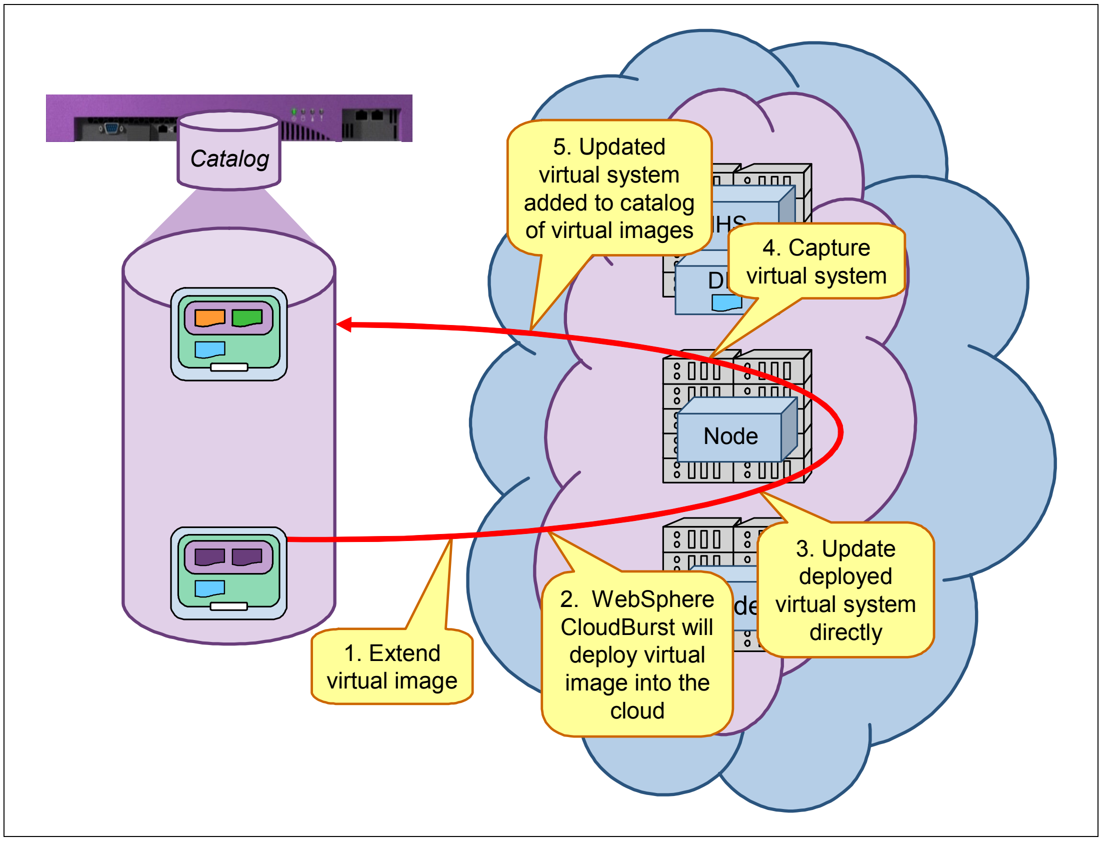
Catalog
5. Updated virtual system added to catalog of virtual images
IHS
DM
4. Capture virtual system
Node
1. Extend virtual image
2. WebSphere CloudBurst will deploy virtual image into the cloud
Node
3. Update deployed virtual system directly

Figure 2 Creating custom images using extend and capture

During the extend and capture process, the appliance creates a running virtual machine from a virtual image that is already included in the catalog. At that point, you can log in, make modifications (for example, install software), and then capture it as a new, custom image in the appliance catalog. With the image in the catalog, the single image is used to make a number of separate patterns and, thus, distinct application environments. WASDTO periodically uses this mechanism to capture major operating system upgrades or patches.

WASDTO also needed a way to apply the most up-to-date patches to the environments that were deployed using WebSphere CloudBurst. In this case, the extend and capture process was not ideal, because WASDTO did not want an unmanageable proliferation of images. Instead, WASDTO pursued a more dynamic, late-binding option. Script packages were created by WASDTO to include in all of the patterns. During the deployment process, these script packages scan the operating system, determine if any patches are necessary, and, if so, download and install the patches. Finally, the script packages register the deployed system with the compliance monitoring server. Figure 3 illustrates this process.

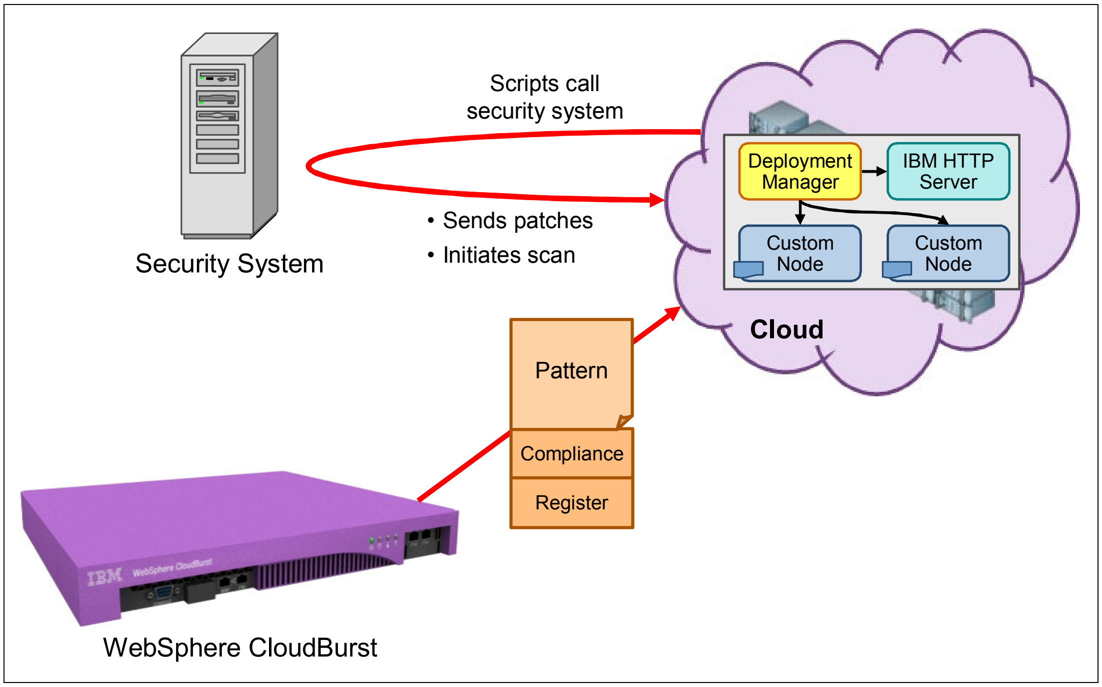

Figure 3 Dynamic patching and registration

This approach enables WASDTO to overcome historical issues with achieving compliance in virtualized environments. WASDTO uses the extend and capture mechanism to create a small set of images containing major operating system patches and upgrades. The team does not create a new image for each operating system patch, and thus the proliferation of virtual images and accompanying management challenges is not an issue.

Each time that the appliance deploys a system based on one of these custom images, two activities occur. First, custom configuration scripts are run to ensure compliancy based on required patches. Second, the scripts register the system to the monitoring server. The result is that each environment deployed using the appliance is compliant and known to the monitoring system. In addition, this process occurs as an automated function of the deployment process, meaning that administrators do not need to apply patches manually or register systems.

In summary, this approach is a significant benefit to WASDTO. The necessary configuration actions to achieve compliancy are an automated part of the deployment process. Therefore, WASDTO does not have to dedicate testers to ensure and enforce compliancy in the lab. In addition, WASDTO no longer has to segregate virtualized WebSphere Application Server environments; thus, testers can access test environments without navigating through numerous firewalls.

Securing access and use of WebSphere CloudBurst

WebSphere CloudBurst, like many cloud computing solutions, promotes a self-service access model to service delivery. This self-service access enables efficiency by eliminating, where appropriate, a multi-step request chain for IT services. However, in an enterprise setting, self-service access is impossible to achieve without effective controls around which users can take the needed actions on specific resources.

WASDTO supports multiple teams and users with WebSphere CloudBurst. These teams vary in terms of responsibilities, levels of expertise in using the product, and deployment goals. To effectively share and promote self-service access among these teams, WASDTO needed a way to classify, categorize, and manage a multitude of users and groups. So, WASDTO defined the users and groups in the appliance, assigned permissions to groups, and controlled the resource access among the groups.

Defining and managing users and groups

Like many organizations, WASDTO already used an enterprise Lightweight Directory Access Protocol (LDAP) server for user authentication. With the WebSphere CloudBurst Appliance, WASDTO integrated data from the existing LDAP server for use in user and group management. When users log in to the appliance, the LDAP server controls authentication. In this way, WASDTO does not have to store and, subsequently, manage passwords in the appliance. In addition, the LDAP server manages user group membership, automatically adjusting user groups in the appliance to match those groups in the LDAP server.

Defining user roles and permissions

The team used LDAP for user authentication and group management; however, the LDAP server did not control authorization. To define user authorization, the team needed to define roles within WebSphere CloudBurst. For ease of management, the team chose to define roles at the user group level, thus allowing the team to define a set of roles that applied to multiple users at one time. WASDTO categorized the following role-based groups:

- *Pattern deployers*: This group has permission to deploy patterns. Typically, these users have less WebSphere Application Server administration expertise and want to deploy constructed, configured environments.
- *Pattern authors and catalog managers*: This group has permission to create patterns, upload script packages, and create custom images. These users are typically seasoned WebSphere Application Server administrators who can build and configure application environments. They simply mapped their existing configuration knowledge to the various customization approaches in WebSphere CloudBurst.
- *Cloud and appliance administrators*: This group has permission to administer the cloud infrastructure and WebSphere CloudBurst. These users are familiar with the configuration and administration of the hardware components within the cloud. In addition, they acquired the skills necessary to manage and maintain the appliance.

Establishing resource access

WASDTO needed to manage access to two WebSphere CloudBurst resources: cloud groups and cloud patterns. To manage this access, WASDTO used the granular access controls in the appliance.

As discussed earlier, cloud groups provide the means to group sets of hypervisors logically into smaller subclouds. For example, an organization might choose to separate the cloud resources that are used for development, test, and quality assurance purposes, as shown in Figure 4.

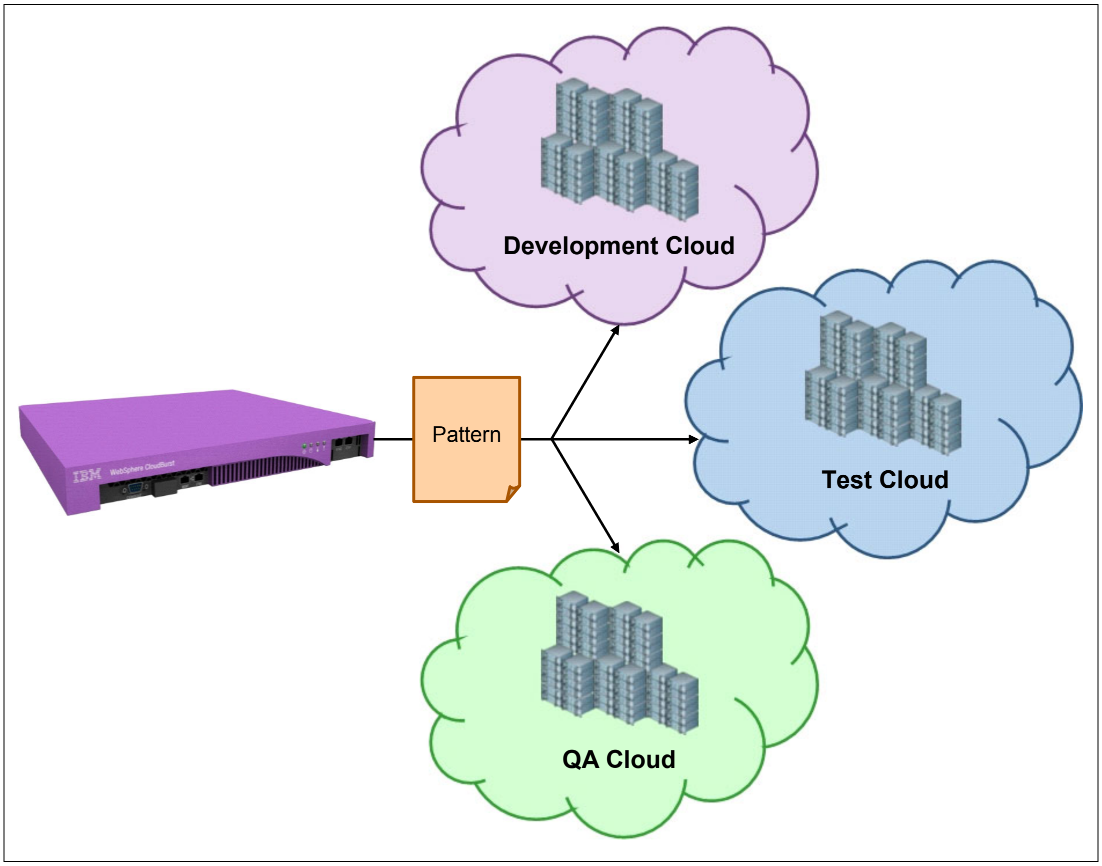

Figure 4 WebSphere CloudBurst cloud groups

Albeit for other purposes than those groups shown in Figure 4, WASDTO established cloud groups to manage resource usage across various teams and efforts. A cloud administrator created a cloud group and decided which resources to put into the group based on its purpose. Then, the cloud administrator decided which users and groups needed access to which cloud groups, thus allowing the team to effectively control which users and groups used which pools of resources. This approach enables the effective sharing of cloud resources, meanwhile accommodating for the special needs of users (for example, performance testers required access to higher-end machines). Further, this approach prevents a single group of users from monopolizing the entire set of cloud resources. Users in a group can consume all of the resources in the cloud group to which they have access, but they cannot consume resources in a cloud group to which they do not have access.

WASDTO applies the same kind of access controls to WebSphere CloudBurst patterns. After creating and verifying a pattern, the pattern author determines which groups need access to the pattern based on the type of environment the pattern represents. When users log in, they only see patterns to which a pattern author has granted them access.

To illustrate the importance of this capability, consider the less experienced group of WebSphere Application Server administrators. If they log in to the appliance and see a list of all patterns, they might not know what to deploy. So, there is no benefit to self-service access. To remedy this situation, the pattern authors in WASDTO take care to expose users to only the necessary patterns. As a result, there is no confusion among users about which patterns to deploy.

Security and compliance

To adopt WebSphere CloudBurst in your organization, you can benefit from the approach that WASDTO used to ensure security. The following steps show the team’s process:

1. Start by identifying the security and compliance requirements that the organization has in place for application environments.
2. Determine how to address those requirements using WebSphere CloudBurst features, such as extend and capture and script packages.
3. Define access to, and use of, the appliance. Determine who the users are, and establish groups based on common roles. Decide which users or groups need access to which resources. For instance, will there be one large cloud that everyone uses, or will there be cloud groups and access controls, similar to WASDTO? (See “Establishing resource access” on page 12.) By effectively using user permissions and granular access controls, you can enable a self-service model using WebSphere CloudBurst.

A customized cloud

Of primary importance among the benefits of a private cloud solution is that it provides and maintains a high degree of control over the elements in the cloud. This degree of control means that you can customize each element of the solution to build a cloud that fits your needs.

A high degree of customization is of paramount importance when considering the creation of a cloud that contains your application environments. After all, you likely do not use the standard default operating systems and middleware. You likely modify the operating system, for example, by installing additional components and making configuration tweaks to suit your needs. At the middleware layer, you install custom applications and configure the runtime to meet your needs. A high degree of customization capability is a requirement when creating a cloud to host your application environments.

Therefore, in setting up the WASDTO cloud-based application environments to support testing efforts, the team needed to customize each layer of the software stack. The appliance provides various avenues of customization, including the following:

- *Extend and capture:* Extend and capture allowed WASDTO to create custom WebSphere Application Server Hypervisor Edition images
- *Script packages:* Script packages allowed WASDTO to automate customization actions as part of the deployment process
- *Deploy-time parameters:* Deploy-time parameters provided the means to customize each pattern deployment

Using extend and capture

Several of the customizations that are required by WASDTO are necessary in each of the environments dispensed by the appliance. WASDTO required customizations, such as WebSphere Application Server monitoring software and operating system enhancements beyond those enhancements included in the WebSphere Application Server Hypervisor Edition. To accomplish these customizations, WASDTO uses the extend and capture mechanism (see "Securing environments in the cloud" on page 10). Using this process, the team installs monitoring software and makes operating system enhancements one time, then captures the customized state of the virtual machine as a new, custom WebSphere Application Server Hypervisor Edition image. For example, WASDTO installs the IBM Tivoli Monitoring Agent using extend and capture. These images become the basis for WebSphere CloudBurst patterns, thus ensuring that all virtual machines created during deployment contain the monitoring software and operating system enhancements.

WASDTO uses the extend and capture mechanism sparingly for two major reasons:

- First, WASDTO wants to keep the inventory of virtual images to a minimum. As discussed in "Forming the cloud" on page 6, this lower inventory reduced cloud storage requirements, and it prevented the kind of image proliferation that presents resource management challenges.
- Second, the extend and capture process requires a full transfer and capture of the virtual image, and it takes about 2 hours to complete. Because of the time involved for several customizations, even those customizations that are necessary in every environment, WASDTO often favors the script package approach. The team makes the trade-off of running a relatively fast script during each deployment, as opposed to spending considerably more time capturing the customization in the image. In addition to saving this time, WASDTO can also modify the script independently of the underlying image. This capability allows them to quickly update the customization action when necessary.

Using script packages

Script packages allow WASDTO to automate the application of customizations during the pattern deployment process. The team creates script packages for a variety of deployment-time customizations, including, but not limited to the following script packages:

- Applying operating system patches (see "Securing environments in the cloud" on page 10)
- Registering with compliance systems (see "Securing environments in the cloud" on page 10)
- Installing applications to the WebSphere Application Server
- Installing the test automation framework and tools
- Registering monitoring agents with the monitoring server

Several script packages, such as those that install applications and test tools, represent customizations that vary across environments. These script packages provide the necessary configuration actions to create a unique, purpose-built testing environment. Therefore, the actual script packages that are used to perform these activities vary based on the target testing environment (for example, performance testing versus functional regression testing).

Other script packages, such as applying operating system patches and registering monitoring agents, are necessary customizations in every environment. For the most part, WASDTO uses these script packages in every environment to apply late-binding changes or those changes for which it makes a trade-off of not creating a custom image (see "Using extend and capture" on page 14). The registration of monitoring agents with a monitoring server illustrates how the team effectively combines customizations achieved using extend and capture with customizations achieved using script packages. The team installs Tivoli Monitoring agents using extend and capture and then creates scripts that configure the agents within the deployed virtual machines to report to the desired monitoring server.

Using deploy-time parameters

With custom patterns built on top of custom images, WASDTO was able to capture a significant majority of the customizations required in the testing environments. However, no matter how much information WASDTO includes in these assets, each deployment of a pattern requires a certain level of unique configuration. WebSphere CloudBurst deploy-time parameters provide WASDTO with the capability to apply these customizations. The definition of deploy-time parameters occurs in two ways:

- The appliance defines a set of standard parameters for parts in a pattern. For example, these standard parameters include WebSphere Application Server cell and node names, passwords, and virtual machine memory and CPU allocation. This set of standard parameters allows the team to shape each deployment in terms of WebSphere Application Server and virtual machine configuration. It is important to note that WASDTO does not want to expose certain parameters, such as root user passwords, to the deployers of a pattern. To prevent this exposure from happening, when creating custom patterns, the team simply locks certain parameters and their values into the pattern. This locking prevents deployers from seeing or changing these parameter values.
- Custom script packages also allow WASDTO to supply deploy-time parameters. For each script package, WASDTO has the ability to define parameters. Like the standard set of parameters, these parameters become a part of the pattern definition. The team uses these parameters for a wide variety of configuration information, such as the location of back-end data sources, the location of monitoring servers, application configuration, and more. As with the standard set of parameters, pattern authors can lock any of these parameter values into the pattern, thus preventing their change during deployment.

Beyond the use of pattern parameters, the test organization also established procedural standards for using deploy-time customizations. The first process was a standard virtual system naming scheme. During the pattern deployment process, deployers provide a name that matches the purpose of the system. For example, a system that is used for regression testing on WebSphere Application Server V7 using an application named DayTrader has the name `WAS70:REGR:daytrader:test`. The team also established scheduling standards. During the deployment process, deployers provide a termination time for their virtual system. This termination time automates the removal of the system and the return of resources to the shared pool at the specified date.

Customizations for your cloud

You probably have the same types of customization requirements as WASDTO. Likely, you will require the ability to install custom software components on top of the operating system, install applications and application resources onto the WebSphere Application Server environment, and make slight tweaks for each deployment to ensure its uniqueness. As you go about identifying the unique customizations that your application environments require, map those unique customizations to a customization capability in the appliance.

You can perform this mapping using several determining factors, including these factors:

- *Prevalence of customization*: For customizations common to most environments, you probably will use extend and capture. However, keep in mind the time that the process consumes and the frequency of applying new customizations. Determine if making a trade-off of a script that runs during every deployment is worthwhile.
- *Target for customization*: As a rule, use script packages for configuring the middleware layer, such as installing applications and configuring application resources.
- *When to apply customization*: Do you need to apply late-binding changes to the environment? This task is occasionally necessary, as illustrated by the WASDTO script

packages that were used to configure monitoring agents and apply the latest operating system patches during deployment.

- *Variability of customization*: Do your customizations vary for each deployment? If so, use parameterized script packages. The script packages provide the configuration logic, but the exact effect depends on deploy-time parameters that are provided by you.

By considering these factors, you can effectively meet customization requirements using features that are built into the appliance. The result will be tailored WebSphere Application Server environments running on a private cloud.

Automation with WebSphere CloudBurst

Many processes in IT can benefit from automation that increases the speed and consistency of execution, whether in cloud computing or traditional service delivery models. Replacing manual and typically tedious tasks with automated processes boosts efficiency in any IT organization. Coupling cloud computing with automation further amplifies the rapid and standardized service delivery promised by this new computing model.

For WASDTO, agility is top priority. The team makes it a point to automate as many processes as possible. Automation provides WASDTO with the means to execute complex, multi-step tasks both quickly and consistently. With the adoption of the appliance, WASDTO benefited from COTS, automated provisioning, and configuration. Beyond that, the team uses the CLI of the appliance to layer additional automation. In particular, WASDTO uses the CLI as a means to automate the WebSphere CloudBurst cloud management and the management of the resources of the appliance, such as virtual images and patterns.

Automated management for cloud infrastructure

Because of constantly changing resource needs, the WASDTO team decided early on in the use of the appliance not to build a fixed size cloud. The team needed the ability to seamlessly move machines to and from the WebSphere CloudBurst cloud to meet objectives across the organization. To create an elastic WebSphere CloudBurst cloud, WASDTO uses both the WebSphere CloudBurst CLI and the Tivoli Provisioning Manager. With this approach, the team can increase and decrease the number of hypervisors in the cloud in an automated manner, based on constantly changing needs.

WASDTO established a set of physical server capacities for use in either native tests or WebSphere CloudBurst tests as resource needs fluctuated. WASDTO utilizes the Tivoli Provisioning Manager to migrate physical servers into the WebSphere CloudBurst cloud, as depicted in Figure 5.

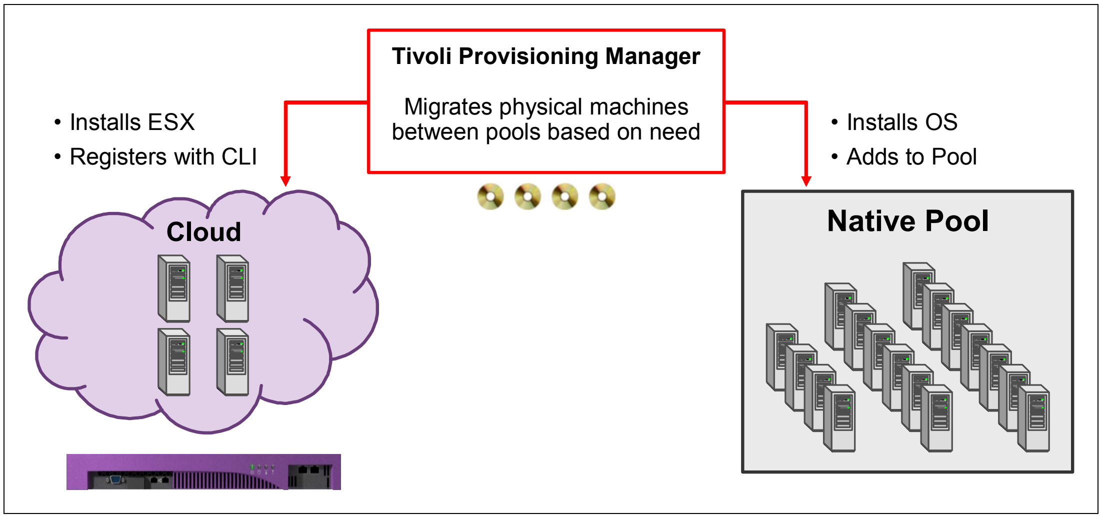
Tivoli Provisioning Manager
Migrates physical machines between pools based on need
• Installs ESX
• Registers with CLI
• Installs OS
• Adds to Pool
Cloud
Native Pool

Figure 5 Managing cloud resources

When additional capacity is required in the WebSphere CloudBurst cloud, the Tivoli Provisioning Manager automates the installation of VMware ESX software on the target server. After the installation of the VMware ESX software, the Tivoli Provisioning Manager workflow calls an automation script that uses the WebSphere CloudBurst CLI to perform the following tasks:

1. Define the new hypervisor resource to the appliance.
2. Add the hypervisor as the sole member of a test cloud group.
3. Deploy a simple pattern to the test cloud group to verify that the hypervisor functions correctly.
4. Move the hypervisor to a production cloud pool if the deployment succeeded, or move the hypervisor to maintenance mode for later debugging by the administrator.

This automated set of actions allows the team to automate the addition of new hypervisors to the cloud. In addition, because the automation conducts test deployments to verify the health of the hypervisor, adding new hypervisors does not compromise the overall state of the cloud.

Of course, the workflow is bidirectional, with the ability to both add and remove capacity from either the WebSphere CloudBurst cloud or the physical resource pool. There is a corresponding automated process that starts by using the WebSphere CloudBurst CLI to remove hypervisor resources from the appliance. At that point, the process invokes the Tivoli Provisioning Manager to remove VMware ESX software from the server, install an operating system, and add it to the native pool of resources.

Automated management of images and patterns

On a daily basis, WASDTO receives new builds of the WebSphere Application Server Hypervisor Edition that the team needs to download and test. To accomplish this download and test in a rapid and consistent manner, the team constructed an automated approach using the WebSphere CloudBurst CLI. The automated process accounts for the following actions:

- Importing a new WebSphere Application Server Hypervisor Edition image into the catalog
- Creating a single server pattern based on the new image and deploying that pattern to prime the cache

- Cloning existing patterns and updating their base image to the new version

To avoid having to watch for builds manually, the team configured agents to monitor a build server feed and listen for each new build of the WebSphere Application Server Hypervisor Edition. When a new build publishes, the automated process begins. A WebSphere CloudBurst CLI script runs to create a virtual image resource by pointing to the location of the new image on the build servers. This script initiates the download of the new image to the target WebSphere CloudBurst Appliance.

With the new WebSphere Application Server Hypervisor Edition in the catalog, a WebSphere CloudBurst CLI script creates a new single server pattern using the image. The team uses these patterns to create a cache for the new virtual images on each possible storage system (see “Automated cloud management” on page 8).

After loading the image and priming the hypervisor cache, the CLI scripts clone existing patterns and base these new patterns on the newest virtual image. Therefore, testers can create test environments that are based on the newest build of the WebSphere Application Server Hypervisor Edition, as shown in Figure 6.

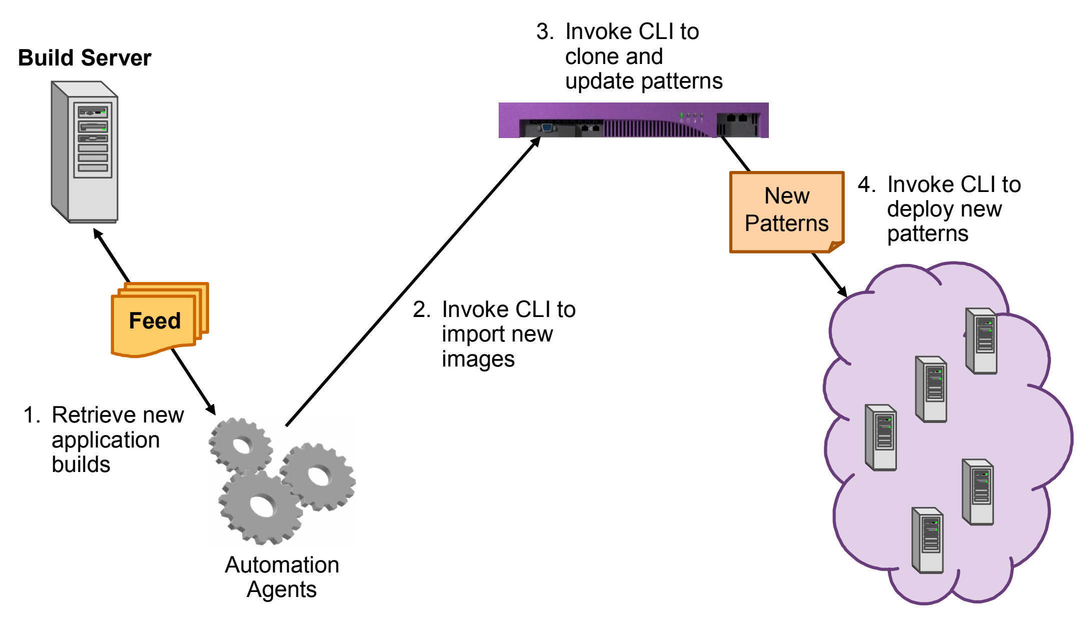
Build Server
3. Invoke CLI to clone and update patterns
New Patterns
4. Invoke CLI to deploy new patterns
Feed
2. Invoke CLI to import new images
1. Retrieve new application builds
Automation Agents

Figure 6 Automation of updated pattern deployment

Automating these steps allows the test organization to codify a process for downloading new builds of the WebSphere Application Server Hypervisor Edition, prime hypervisor caches for the new image, and update patterns used for testing environments based on the new build of the virtual image. This automation results in a faster, more repeatable method for deploying test environments for daily WebSphere Application Server Hypervisor Edition builds.

Planning your automation techniques

The needs of WASDTO might differ from your needs. After all, it is unlikely that you need to handle daily updates of the WebSphere Application Server Hypervisor Edition images. However, you might need frequent updates to your applications, periodic updates to your WebSphere Application Server software, the ability to fluctuate cloud capacity, or other recurring activities that benefit from automation. Therefore, you can extrapolate the work performed in WASDTO and apply it to your automation approach with WebSphere CloudBurst.

The first step is to identify the continual, repeating processes that you perform with the appliance. These processes might include the deployment of patterns, expansion and contraction of cloud resources, updating of deployed systems, or any number of other processes. With these continual, repeating processes identified, make a plan to automate those processes. You can use the WebSphere CloudBurst CLI, or even the REST API, to leverage the same appliance functionality that you utilize in the web console. If you already have an automated workflow solution, such as Tivoli Provisioning Manager, you can integrate WebSphere CloudBurst automation into your broader management processes. By constructing automated approaches to these repeated processes, you can extend the speed and consistency that you derive from the product.

The value of WebSphere CloudBurst

A meaningful part of adopting a new technology is establishing a means to measure the value that it provides to your organization. Measuring and communicating the value of a particular solution is important in either justifying continuing investment or proving that continuing investment is unwarranted. Cloud computing typically requires process changes that span multiple organizations. Therefore, each organization must see direct value for its initial and continuing investment.

In the adoption and implementation of the appliance, WASDTO constructed a plan for both measuring and communicating the value of the appliance.

Measuring and communicating the value of WebSphere CloudBurst

To measure the value of the appliance, WASDTO established a plan consisting of three important activities:

- Identify value-centric metrics
- Measure metrics for the traditional provisioning approach
- Measure metrics for the WebSphere CloudBurst provisioning approach

The first step was to identify which metrics to measure. The team members considered metrics that quantified their progress on achieving their goals of improving availability, utilization, and manageability for their WebSphere Application Server testing environments. As a result, the team identified the following metrics:

- *Hardware utilization*: The team members measured the average utilization of the hardware resources in both their native pool of resource and their WebSphere CloudBurst cloud.
- *Time to deploy new environments*: The team measured the end-to-end deployment time from the installation of the operating system, to the installation and configuration of WebSphere Application Server for both traditional deployments and those deployments that were performed by the appliance.
- *Percentage of failed or corrupt deployments*: The team calculated the ratio of deployments that resulted in unusable environments for both traditional deployments and those deployments that were performed by the appliance.
- *Number of non-compliant environments*: The team calculated the ratio of deployments that resulted in non-compliant environments for both traditional deployments and those deployments that were performed by the appliance.

- *User satisfaction*: The team conducted user surveys to determine the percentage of users that preferred WebSphere CloudBurst to traditional provisioning techniques.

The team conducted these measurements on an ongoing basis. At the end of the first year of adoption, the team reported the following results:

- *Hardware utilization*: With the appliance, hardware utilization consistently averaged between 60-70%. This utilization was a significant improvement over the 6-12% average hardware utilization rate in the test organization's native resource pool.
- *Time to deploy new environments*: Deployment time with WebSphere CloudBurst ranged between 15-30 minutes depending on the pattern selected for deployment. The team's traditional provisioning approach experienced deployment times that averaged 3 hours.
- *Percentage of failed or corrupt deployments*: Before adopting the appliance, the team deemed between 20-50% of deployments as failed or corrupt. Using the appliance, the team classified only 5% of deployments as failed or corrupt. The majority of the failed deployments occurred solely because the team ran out of capacity in the WebSphere CloudBurst cloud.
- *Number of non-compliant environments*: Traditionally, the test organization dedicated at least one person to update and patch non-compliant deployments. With the appliance, patterns and script packages encapsulated the update and patch process for operating system compliance. The result of this automated approach toward compliance assuredness is that the team deploys zero non-compliant environments. Before adopting this approach, the team had to dedicate testers each day to manually update environments for compliance reasons.
- *User satisfaction*: The test organization conducted user surveys to determine the percentage of testers that preferred WebSphere CloudBurst to traditional provisioning techniques. The result was that 80% of the users preferred to use WebSphere CloudBurst to provision their environments as opposed to traditional provisioning techniques.

In addition to these types of measurements, the team measured the bottom-line effect to the organization by performing a return on investment (ROI) study of their incremental WebSphere CloudBurst adoption. WASDTO found that in the first year of adoption alone, it achieved direct savings of USD500,000. In addition, the team reported an additional USD2.1 million in enabled efficiency gains supported by an 85% increase in system administrator efficiency.

Again, these results reflect measurements observed after a year of adopting the appliance, during which time the team used the appliance to manage about 6% of its lab infrastructure. It is important to note that the test organization continually collects these measurements. In addition to simply collecting these measurements, the team sets up regularly occurring meetings with those individuals that the team identified as key stakeholders and decision makers within the organization. These meetings serve as an opportunity for the team to report its results and illustrate the value from its incremental adoption approach. By providing empirical and cost savings data, the team secures ongoing commitment and investment in its adoption of the WebSphere CloudBurst Appliance.

Your plan for measuring and communicating value

As WASDTO demonstrates, it is important to formulate and execute a plan for measuring and communicating the value of adopting WebSphere CloudBurst. First, you need to establish the goals for the organization, and you need to identify measurable metrics that indicate your progress toward those goals. It is likely that you can at least start with the same metrics identified by the test organization. From there, you need to establish a plan to measure these metrics for both your WebSphere CloudBurst Appliance and traditional provisioning approaches.

Beyond simply measuring these metrics, identify key decision makers in the organization. Which people or groups of people can influence the expansion of the appliance in the organization? As WASDTO did, consider those people with the resources to drive further adoption in addition to those individuals who can act as evangelists and spread adoption throughout the organization. If you couple the measurement of value with the communication of value, you can effectively influence the further adoption of WebSphere CloudBurst in your organization.

Conclusion

WASDTO has achieved significant results through the incremental adoption of private cloud computing using WebSphere CloudBurst. The team started by identifying a variety of usage scenarios for the cloud, beginning with integration into existing provisioning processes, then providing self-service access to novice users, followed by access for expert WebSphere Application Server administrators. Similarly, the team staged the size of the cloud, increasing capacity over time, and automated the movement of capacity between physical and virtual environments. WASDTO used the customization capabilities of the appliance to implement security policies and to provide repeatable, customized deployment for cloud users. Throughout the process, the team measured key metrics to assess and communicate the results. This incremental, staged adoption led to achieving the objectives for improved availability, utilization, and manageability. These techniques can provide a framework for your own adoption plan.

The authors

Dustin Amrhein joined IBM as a member of the development team for WebSphere Application Server. While in that position, Dustin worked primarily on web services infrastructure and web services programming models. In addition, Dustin worked on the development of a RESTful services framework for Java™ systems. In his current role, Dustin is a technical evangelist for emerging technologies in the IBM WebSphere portfolio, with a specific focus on cloud computing.

Ruth Willenborg is a Senior Technical Staff Member in the IBM WebSphere Technology Institute. Ruth currently works on WebSphere cloud computing and virtual appliance initiatives and is the technical evangelist for the new IBM WebSphere CloudBurst Appliance. Prior to her work on virtualization and appliance initiatives, Ruth was the manager of the WebSphere Performance team, responsible for WebSphere CloudBurst performance analysis, performance benchmarking, and performance tool development. Ruth has over 20 years of experience in software development at IBM. She is the co-author of *Performance Analysis for Java Web Sites* and numerous articles on both WebSphere performance and the use of WebSphere with virtualization technologies.

The authors would like to acknowledge Robbie Minshall for his contributions to this paper. Robbie is an architect in the WebSphere Application Server Development and Test Organization, and his input was instrumental in the authoring of this paper.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

- Ruth Willenborg, et al., *Performance Analysis for Java Web Sites,* Addison-Wesley Professional, 2002, ISBN: 0201844540

IBM Redbooks publications

The following IBM Redbooks® publications provide additional information about the topic in this document. Note that several publications referenced in this list might be available in softcopy only.

- *Rapid WebSphere Application Server Provisioning with WebSphere CloudBurst Appliance*, REDP-4565
- *WebSphere Cloudburst Appliance and PowerVM*, SG24-7806

You can search for, view, or download IBM Redbooks publications, Redpapers™, Technotes or webdocs, draft publications and Additional materials, and order hardcopy IBM Redbooks publications, at this website:

http://ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- WebSphere CloudBurst on developerWorks®

  http://www.ibm.com/developerworks/search/searchResults.jsp?searchType=1&searchSite=dW&searchScope=dW&query=websphere+cloudburst&Search=Search

- WebSphere CloudBurst blog

  https://www.ibm.com/developerworks/mydeveloperworks/blogs/cloudview/?lang=en

- WebSphere CloudBurst information center

  http://publib.boulder.ibm.com/infocenter/wscloudb/v2r0/index.jsp?topic=/com.ibm.websphere.cloudburst.doc/welcome.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4708-00 was created or updated on December 14, 2010.

Send us your comments in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CloudBurst™
developerWorks®
IBM®
PowerVM™
Redbooks®
Redpaper™
Redpapers™
Redbooks (logo) ®
Tivoli®
WebSphere®
z/VM®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.